



Two-point boundary value problems by the extended Adomian decomposition method

Bongsoo Jang^{*,1}

Department of Mathematical Sciences, Kent State University, Ashtabula, OH 44004, USA

Received 4 April 2007; received in revised form 12 July 2007

Abstract

In this paper, we present an efficient numerical algorithm for solving two-point linear and nonlinear boundary value problems, which is based on the Adomian decomposition method (ADM), namely, the extended ADM (EADM). The proposed method is examined by comparing the results with other methods. Numerical results show that the proposed method is much more efficient and accurate than other methods with less computational work.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Two-point boundary value problem; Adomian decomposition method; Shooting methods

1. Introduction

In this paper, we study two-point boundary value problems of the form

$$u'' = f(x, u, u'), \quad a < x < b, \quad (1)$$

subject to the boundary conditions

$$u(a) = \alpha, \quad u(b) = \beta, \quad (2)$$

where f is continuous on the set $D = \{(x, u, u') | a \leq x \leq b, u, u' \in \mathbb{R}\}$.

Two-point boundary value problems have been investigated in many application areas. The most common numerical method for solving these problems is to use shooting methods [6,10]. Although shooting methods have many advantages such as a fast solver and a reduced size of system, it also requires a huge amount of computational work in obtaining accurate approximations especially for nonlinear problems.

The Adomian decomposition method (ADM) has been studied by many scientists [1–3,7–9,12] for solving differential and integral problems in many scientific applications. It decomposes the solution into the series which converges rapidly.

* Tel.: +1 440 9644281; fax: +1 440 9644269.

E-mail addresses: bjang1@kent.edu, bsjang@postech.ac.kr (B. Jang).

¹ Present address: BK21 CoDiMaRO, Department of Mathematics, Pohang University of Science and Technology, Pohang City 790-784, Republic of Korea.

Each component can be easily determined by using a simple recursive relation. Let us rewrite the model problem (1) in operator form as follows:

$$Lu = Nu + \phi, \quad (3)$$

where L is the second-order derivative operator and N is the nonlinear operator that can be defined by $N = \hat{f}$, where $\hat{f}(x, u, u') = f(x, u, u') + \phi(x)$. Applying the inverse operator L^{-1} to both sides of (3), and using the boundary (or initial) condition, we obtain

$$u = g + L^{-1}\phi + L^{-1}Nu, \quad (4)$$

where g represents the term arising from the given boundary (or initial) condition. The standard ADM defines the solution u by the series

$$u = \lim_{n \rightarrow \infty} S_n, \quad S_n = \sum_{i=1}^n u_i,$$

where each component u_i can be determined recursively as follows:

$$u_0 = g + L^{-1}\phi, \quad u_{i+1} = L^{-1}(Nu_i), \quad i \geq 0. \quad (5)$$

It is well known [1] that the nonlinear function $N(u)$ is usually represented by the infinite series of polynomials called Adomian polynomials A_n

$$N(u) = \sum_{n=0}^{\infty} A_n(u_0, u_1, \dots, u_n), \quad (6)$$

where Adomian polynomials A_n are defined by

$$A_n(u_0, \dots, u_n) = \frac{1}{n!} \left[\frac{d^n}{d\lambda^n} N \left(\sum_{i=0}^n u_i \lambda^i \right) \right]_{\lambda=0}. \quad (7)$$

As seen in (4), ADM is based on finding the solution in operator form by taking a suitable inverse operator L^{-1} . Since the operator L is the second-order differential operator, the inverse operator L^{-1} is either twofold definite or indefinite integral.

Let us consider the inverse operator L^{-1} as the twofold definite integral defined by

$$L^{-1} = \int_a^x dx' \int_a^{x'} dx''. \quad (8)$$

This implies

$$L^{-1}Lu = u(x) - u(a) - (x-a)u'(a).$$

Thus, the solution in (3) can be written as

$$u = u(a) + (x-a)u'(a) + L^{-1}\phi + L^{-1}Nu. \quad (9)$$

Applying the standard ADM yields the following recursive scheme:

$$u_0 = u(a) + (x-a)u'(a) + L^{-1}\phi, \quad u_{n+1} = L^{-1}Nu_n, \quad n \geq 0.$$

In order to determine all other components u_n , $n \geq 1$, the zeroth component u_0 has to be determined. However, $u'(a)$ is not defined by the boundary condition so that the zeroth component cannot be directly determined.

Many authors [2,3,7,12] have proposed modified ADMs to overcome this difficulty. In [7,12], $u'(a)$ is set to be a constant, $u'(a) = c$, and it can be determined such that the n th partial sum $S_n(x, c)$ satisfies the boundary condition at

$x = b$ because $S_n(a, c) = u(a)$. In this case, it requires additional computational work to solve the nonlinear equation $S_n(b, c) = u(b)$ for c and the solution c may not be uniquely determined.

In [3] a specific type of (1) is considered as follows:

$$u'' + pu' = f(x, u). \quad (10)$$

By using the inverse operator that is defined by

$$L^{-1} = \int_a^x dx' \int_b^{x'} dx'',$$

and setting $u'(b)$ by a series, $u'(b) = \sum_{n=0}^{\infty} c_n$, each component u_n can be obtained as follows:

$$u_0 = u(a) + (x - a)[c_0 + u(b)p(b)], \quad (11)$$

$$u_{n+1} = c_{n+1}(x - a) - \int_a^x pu_n + L^{-1}p'u_n + L^{-1}f(x, u_n), \quad n \geq 0. \quad (12)$$

In order to determine the unknown constants c_n it is also required that the n th partial sum S_n satisfies the boundary conditions. It is obvious that $S_n(a) = u(a)$. Thus, the following scheme has been proposed to satisfy the boundary condition at $x = b$

$$u_0(b) = u(b), \quad u_n(b) = 0, \quad n \geq 1. \quad (13)$$

However, since each constant c_n , $n \geq 1$, can be determined by solving (13), it also requires additional computational work.

If the inverse operator is defined by the indefinite integral, the standard ADM yields the following recursive scheme [2]:

$$u_0 = c_0^0 + c_1^0 x + L^{-1}\phi, \quad u_{n+1} = c_0^{n+1} + c_1^{n+1} x + L^{-1}A_n, \quad n \geq 0.$$

Unknown constants c_0^n, c_1^n can be determined by satisfying the following conditions: $S_n(a) = u(a)$, $S_n(b) = u(b)$ for each n . However, this approach requires even more computational work compared with other approaches using the definite integral as the inverse operator.

In this work, a new modification of the ADM is proposed to overcome difficulties occurred in the standard ADM for solving two-point boundary value problems, namely, the extended ADM (EADM). Main idea of the EADM is to create a canonical form containing all boundary conditions so that the zeroth component is explicitly determined without additional calculations and all other components are also easily determined.

This paper is organized as follows: in Section 2, the proposed method is analyzed. Several numerical illustrations are demonstrated and results obtained by EADM and other methods are presented in Section 3. Conclusions are given in Section 4.

2. Analysis of the EADM

As described before, the standard ADM by using either a definite or indefinite integral as the inverse operator does require additional computational work in determining each component u_n . It is easy to see that these difficulties originate from the fact that all canonical forms contain the unknown constant. Thus, our main goal is to create a new canonical form containing all boundary conditions so that each component u_n in the recursive scheme can be explicitly determined without additional computational work. Here we define the inverse operator L^{-1} by the twofold definite integral as follows:

$$L^{-1} = \int_a^x dx' \int_b^{x'} dx''. \quad (14)$$

Let us take the inverse operator L^{-1} to (3), then we have

$$u = u(a) + (x - a)u'(b) + L^{-1}\phi + L^{-1}Nu. \quad (15)$$

In what follows we describe the basic idea of the EADM.

Step 1: In order to express $u'(b)$ in terms of the known data, the boundary condition at $x = b$ is applied to (15). Thus we have

$$u(b) = u(a) + (b - a)u'(b) + [L^{-1}\phi + L^{-1}Nu]_{x=b}.$$

Solving for $u'(b)$ yields

$$u'(b) = \frac{u(b) - u(a)}{b - a} - \frac{1}{b - a}[L^{-1}\phi + L^{-1}Nu]_{x=b}. \quad (16)$$

By substituting (16) into (15), we have

$$u = u(a) + q(x)[u(b) - u(a)] - q(x)[L^{-1}\phi + L^{-1}Nu]_{x=b} + L^{-1}\phi + L^{-1}Nu, \quad (17)$$

where $q(x) = (x - a)/(b - a)$.

Step 2: From (17) combined with the boundary conditions, we propose the following recursive scheme:

$$u_0 = \alpha + q(x)(\beta - \alpha) + L^{-1}\phi - q(x)[L^{-1}\phi]_{x=b}, \quad (18)$$

$$u_{n+1} = L^{-1}A_n - q(x)[L^{-1}A_n]_{x=b}, \quad n \geq 0, \quad (19)$$

where A_n is the Adomian polynomials associated with the nonlinear operator N . It is worth noting that the canonical form (17) consists of all boundary conditions. Moreover, the n th partial sum S_n from the recursive schemes, (18) and (19), always satisfies the boundary conditions for any n . Thus, it is not necessary to determine the unknown constant $u'(b)$ by extra calculations.

Remark 1. Let us consider the problem (3) with the scheme in [3], then we have the following recursive relation:

$$u_0 = u(a) + (x - a)c_0 + L^{-1}\phi, \quad u_{n+1} = (x - a)c_{n+1} + L^{-1}A_n, \quad n \geq 0. \quad (20)$$

Applying (13) to u_n , $n \geq 0$ in (20) yields

$$c_0 = \frac{1}{b - a}\{u(b) - u(a) - [L^{-1}\phi]_{x=b}\}, \quad c_{n+1} = -\frac{1}{b - a}[L^{-1}A_n]_{x=b}.$$

This implies that each component u_n is identical to the components u_n in (18) and (19). Thus, our approach is the generalized version in [3] for solving two-point boundary value problems.

Let us recall that each component u_n is determined by the given twofold definite integral as the inverse operator. Since there are several types of twofold definite integrals, it is possible to produce different components for each twofold definite integral. Thus, it is natural to ask a question how different components will be produced in EADM for each twofold definite integral and what is the most appropriate inverse operator in EADM. The answers for these questions are as follows.

To achieve the objectives of the above questions, let us consider the inverse operator L_k^{-1} defined by

$$L_k^{-1} = \int_{v_k}^x dx' \int_{w_k}^{x'} dx'', \quad k = 1, 2, 3, 4, \quad (21)$$

where $\{v_1, v_2, v_3, v_4\} = \{a, a, b, b\}$ and $\{w_1, w_2, w_3, w_4\} = \{b, a, a, b\}$. Let us define u_n^k by the component induced by L_k^{-1} in EADM. Applying the procedures in EADM with the inverse operator L_2^{-1} yields the following recursive scheme:

$$u_0^2 = \alpha + q(x)(\beta - \alpha) + L_2^{-1}\phi - q(x)[L_2^{-1}\phi]_{x=b}, \quad (22)$$

$$u_{n+1}^2 = L_2^{-1}A_n - q(x)[L_2^{-1}A_n]_{x=b}, \quad n \geq 0. \quad (23)$$

Lemma 2. $L_1^{-1}\psi - q(x)[L_1^{-1}\psi]_{x=b} = L_2^{-1}\psi - q(x)[L_2^{-1}\psi]_{x=b}$.

Proof. Let us define ψ_1 and ψ_2 by $(\psi_1)' = \psi$ and $(\psi_2)' = \psi_1$. Then we have

$$L_1^{-1}\psi = \psi_2(x) - \psi_2(a) - (x-a)\psi_1(b), \quad L_2^{-1}\psi = \psi_2(x) - \psi_2(a) - (x-a)\psi_1(a).$$

Therefore, this yields the following:

$$\begin{aligned} L_1^{-1}\psi - q(x)[L_1^{-1}\psi]_{x=b} &= \psi_2(x) - \psi_2(a) - (x-a)\psi_1(b) - \frac{x-a}{b-a}[\psi_2(b) - \psi_2(a) - (b-a)\psi_1(b)] \\ &= \psi_2(x) - \psi_2(a) - q(x)(\psi_2(b) - \psi_2(a)). \end{aligned}$$

In the same manner, we have

$$L_2^{-1}\psi - q(x)[L_2^{-1}\psi]_{x=b} = \psi_2(x) - \psi_2(a) - q(x)(\psi_2(b) - \psi_2(a)). \quad \square$$

Corollary 3. From Lemma 2, it is easy to see that $u_n^1 = u_n^2, n \geq 0$.

Now let us consider each component u_n^3 by taking L_3^{-1} as the inverse operator. Each component u_n^3 can be easily obtained by the same procedures in EADM combined with the boundary condition at $x = a$ as follows:

$$\begin{aligned} u_0^3 &= \beta + r(x)(\alpha - \beta) + L_3^{-1}\phi - r(x)[L_3^{-1}\phi]_{x=a}, \\ u_{n+1}^3 &= L_3^{-1}A_n - r(x)[L_3^{-1}A_n]_{x=a}, \quad n \geq 0, \end{aligned}$$

where $r(x) = (x-b)/(a-b)$.

Lemma 4. $u_n^1 = u_n^3, n \geq 0$.

Proof.

$$1 - r(x) = 1 - \frac{x-b}{a-b} = \frac{a-x}{a-b} = q(x). \quad (24)$$

This implies that $\beta + r(x)(\alpha - \beta) = \beta + (1 - q(x))(\alpha - \beta) = \alpha + q(x)(\beta - \alpha)$. Thus, it is sufficient to show that $L_1^{-1}\psi - q(x)[L_1^{-1}\psi]_{x=b} = L_3^{-1}\psi - r(x)[L_3^{-1}\psi]_{x=a}$. Let us denote ψ_1 and ψ_2 by $(\psi_1)' = \psi$ and $(\psi_2)' = \psi_1$. Then, we have

$$\begin{aligned} L_1^{-1}\psi - q(x)[L_1^{-1}\psi]_{x=b} &= \psi_2(x) - \psi_2(a) - (1 - r(x))(\psi_2(b) - \psi_2(a)) \\ &= \psi_2(x) - \psi_2(b) - r(x)(\psi_2(a) - \psi_2(b)) \\ &= L_3^{-1}\psi - r(x)[L_3^{-1}\psi]_{x=a}. \end{aligned}$$

It completes the proof. \square

In a similar manner in Lemma 2, it is easy to show that $u_n^3 = u_n^4, n \geq 0$. Thus, the following conclusion can be obtained.

Theorem 5. Every component $u_n^k, n \geq 0$ induced by $L_k^{-1}, k = 1, 2, 3, 4$ in EADM is identical. In other words, EADM is independent on the inverse operator which is defined by any twofold definite integral.

3. Examples

In this section, we demonstrate the effectiveness of the EADM with several illustrative examples. To do that, for each example, the maximum error for the n th partial sum S_n is compared with the exact solution. Results are depicted in log–log scale. Moreover, all numerical results obtained by EADM are compared with the results obtained by various numerical methods.

Table 1
Absolute errors for Example 1

x	$ u - S_7 $ in [3]	$ u - S_7 $ in [12]	$ u - S_7 $ in EADM
1.1	$4.23 \cdot 10^{-4}$	$6.40 \cdot 10^{-9}$	$8.00 \cdot 10^{-10}$
1.2	$6.63 \cdot 10^{-4}$	$1.33 \cdot 10^{-8}$	$7.00 \cdot 10^{-10}$
1.3	$6.41 \cdot 10^{-4}$	$2.08 \cdot 10^{-8}$	$3.81 \cdot 10^{-9}$
1.4	$4.26 \cdot 10^{-4}$	$2.86 \cdot 10^{-8}$	$6.82 \cdot 10^{-9}$
1.5	$1.44 \cdot 10^{-4}$	$3.65 \cdot 10^{-8}$	$8.74 \cdot 10^{-9}$
1.6	$9.72 \cdot 10^{-5}$	$4.34 \cdot 10^{-8}$	$8.89 \cdot 10^{-9}$
1.7	$2.34 \cdot 10^{-4}$	$4.76 \cdot 10^{-8}$	$7.50 \cdot 10^{-9}$
1.8	$2.49 \cdot 10^{-4}$	$4.58 \cdot 10^{-8}$	$4.99 \cdot 10^{-9}$
1.9	$1.59 \cdot 10^{-4}$	$3.25 \cdot 10^{-8}$	$2.32 \cdot 10^{-9}$

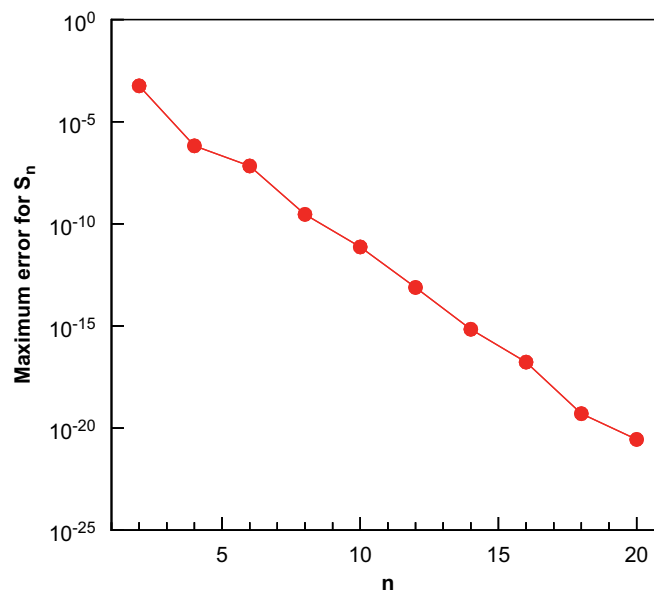


Fig. 1. Maximum error $\|u - S_n\|_\infty$ for Example 1.

Example 1. Let us consider the following linear problem [3]:

$$x^2 u'' - x u' + u = 0, \quad 1 < x < 2,$$

subject to the boundary conditions

$$u(1) = 1, \quad u(2) = 1.$$

It is easy to see that the exact solution is $u(x) = x - \frac{x \ln x}{2 \ln 2}$. By dividing by x^2 , we have

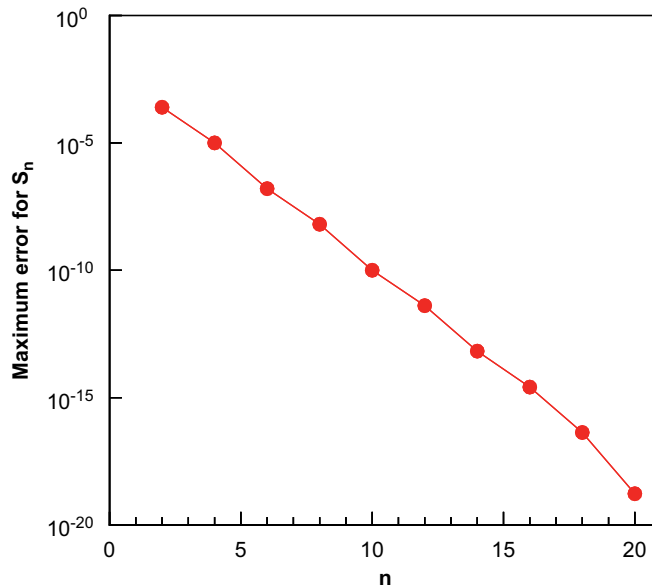
$$u'' = \frac{1}{x} u' - \frac{1}{x^2} u.$$

From the recursive schemes (18) and (19), each component u_n can be easily obtained. All numerical results are compared with the results obtained by the modified ADMs in [3,12]. Table 1 shows the absolute error at each test point between the exact solution and the seventh partial sum S_7 . This shows that EADM is much more efficient than the other modified ADMs in [3,12]. It is worth noting that even though EADM is the generalized version of the modified ADM in [3], it may yield a different numerical approximation because $L^{-1}(pu')$ in [3] is completely expanded; $L^{-1}(pu') = -(x-a)p(b)u(b) + \int_a^x pu - L^{-1}(p'u)$ so that each component is influenced by the function p , whereas $L^{-1}(pu')$ in EADM is not expanded.

Table 2

Maximum errors for Example 2: mesh size $h = 0.01$ in [4,5]

Methods	Maximum error
Finite element method in [5]	$6.36 \cdot 10^{-7}$
Finite volume method in [5]	$3.18 \cdot 10^{-7}$
B-spline method in [4]	$2.89 \cdot 10^{-10}$
S_{10} in [12]	$7.65 \cdot 10^{-10}$
S_{10} in EADM	$1.05 \cdot 10^{-10}$

Fig. 2. Maximum error $\|u - S_n\|_\infty$ for Example 2.

Maximum errors of S_n , $n \leq 20$, are depicted in Fig. 1. It shows that the n th partial sum S_n in EADM converges exponentially to the exact solution as n grows. It is worth noting that only two iterations are required to obtain the same order of the maximum error in [3] that used seven iterations. In detail, $\|u - S_2\|_\infty = 5.7 \cdot 10^{-4}$ in EADM and $\|u - S_7\|_\infty = 6.8 \cdot 10^{-4}$ in [3].

Example 2. Let us consider the following linear problem [4,5]

$$u'' = u' - \exp(x - 1) - 1, \quad 0 < x < 1,$$

subject to the boundary conditions

$$u(0) = 0, \quad u(1) = 0.$$

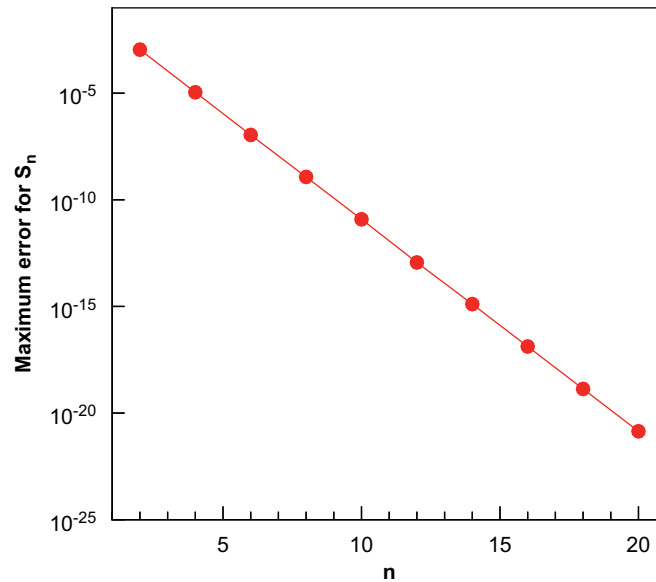
The exact solution is $u(x) = x(1 - \exp(x - 1))$.

Maximum errors obtained by various methods in [4,5], the modified ADM in [12] and the EADM are shown in Table 2. It shows that the EADM with ten iterations S_{10} gives a similar maximum error as the B-spline method. Let us note that each component u_n can be easily computed by any symbolic packages so that EADM with a few more iterations provides much more accurate approximation without solving a system as in [4,5]. Convergence rate of $\|u - S_n\|_\infty$, $n \leq 20$, is shown in Fig. 2.

Table 3

Absolute errors for Example 3: mesh size $h = 1/8$ in [11]

x	EFM in [11]	$ u - S_5 $ in [12]	$ u - S_5 $ in EADM
1/8	$6.88 \cdot 10^{-5}$	$2.09 \cdot 10^{-10}$	$4.37 \cdot 10^{-7}$
2/8	$4.93 \cdot 10^{-5}$	$4.22 \cdot 10^{-10}$	$8.07 \cdot 10^{-7}$
3/8	$3.21 \cdot 10^{-5}$	$6.42 \cdot 10^{-10}$	$1.05 \cdot 10^{-6}$
4/8	$2.63 \cdot 10^{-5}$	$8.72 \cdot 10^{-10}$	$1.14 \cdot 10^{-6}$
5/8	$2.16 \cdot 10^{-5}$	$1.10 \cdot 10^{-9}$	$1.05 \cdot 10^{-6}$
6/8	$1.09 \cdot 10^{-5}$	$1.31 \cdot 10^{-9}$	$8.07 \cdot 10^{-7}$
7/8	$1.01 \cdot 10^{-5}$	$1.26 \cdot 10^{-9}$	$4.37 \cdot 10^{-7}$

Fig. 3. Maximum error $\|u - S_n\|_\infty$ for Example 3.

Example 3. Let us consider the following linear problem [11]:

$$u'' = u + \cos(x), \quad 0 < x < 1,$$

subject to the boundary conditions

$$u(0) = 1, \quad u(1) = 1.$$

The exact solution is $u(x) = c_1 \exp(x) + c_2 \exp(-x) - \cos(x)/2$, where

$$c_1 = \frac{-3 \cosh(1) + 3 \sinh(1) + \cos(1) + 2}{4 \sinh(1)}, \quad c_2 = \frac{3 \cosh(1) + 3 \sinh(1) - \cos(1) - 2}{4 \sinh(1)}.$$

Absolute error at each test point between the exact solution and the results obtained by the exponential fitting method (EFM) in [11], the modified ADM in [12] and the EADM is compared in Table 3. With only five iterations a better approximation S_5 has been obtained than the results by EFM. Even though the modified ADM shows a better performance than EADM, it is easy to obtain a similar accurate approximation with a few more iterations in EADM. Convergence rate for S_n , $n \leq 20$, is depicted in Fig. 3.

Example 4. Let us consider the following nonlinear problem [6]:

$$u'' = u^2 + 2\pi^2 \cos(2\pi x) - \sin^4(\pi x), \quad 0 < x < 1,$$

Table 4
Absolute errors for Example 4

x	Shooting method	$ u - S_7 $ in EADM
0.1	$0.8 \cdot 10^{-6}$	$6.9 \cdot 10^{-7}$
0.2	$2.8 \cdot 10^{-6}$	$1.3 \cdot 10^{-6}$
0.3	$5.4 \cdot 10^{-6}$	$1.9 \cdot 10^{-6}$
0.4	$7.5 \cdot 10^{-6}$	$2.3 \cdot 10^{-6}$
0.5	$8.3 \cdot 10^{-6}$	$2.5 \cdot 10^{-6}$
0.6	$7.5 \cdot 10^{-6}$	$2.3 \cdot 10^{-6}$
0.7	$5.4 \cdot 10^{-6}$	$1.9 \cdot 10^{-6}$
0.8	$2.7 \cdot 10^{-6}$	$1.3 \cdot 10^{-6}$
0.9	$0.6 \cdot 10^{-6}$	$6.9 \cdot 10^{-7}$

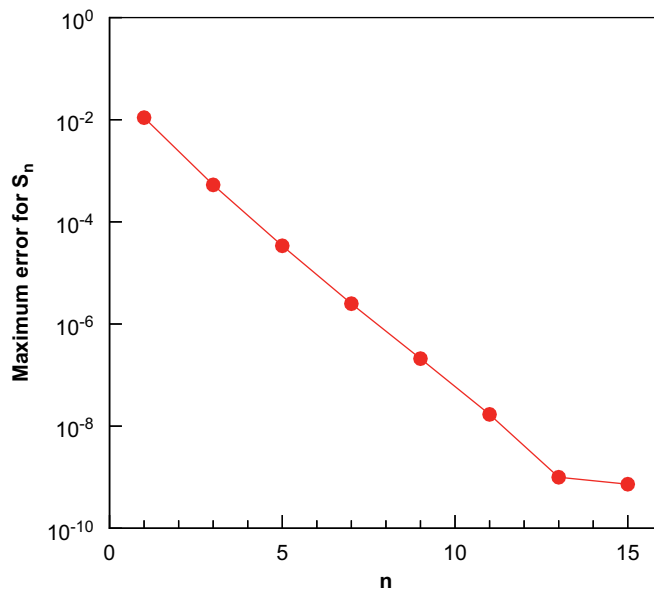


Fig. 4. Maximum error $\|u - S_n\|_\infty$ for Example 4.

with the boundary conditions

$$u(0) = 0, \quad u(1) = 0.$$

The exact solution is $u(x) = \sin^2(\pi x)$.

For each test point, the absolute error between the exact solution and the results obtained by the shooting method [6] and the EADM is compared in Table 4. It is worth noting that the shooting method requires several numerical procedures in obtaining approximations. In details, the fourth-order Runge–Kutta method (mesh size $h = 0.05$) and Newton's method (error bound $= 10^{-7}$) have been employed. However, the proposed method provides a direct calculation of each component u_n through the recursive scheme by using any symbolic packages. With seven iterations, S_7 gives better numerical approximations than results by the shooting method. Convergence rate for S_n , $n \leq 15$, is depicted in Fig. 4.

4. Conclusions

ADM has been successful for solving many application problems with simple calculations. However, it has difficulties in dealing with boundary conditions for solving two-point boundary problems. Many approaches have been presented to overcome these difficulties. However, they require additional computational work since all boundary conditions are not included in the canonical form. Our fundamental goal is to create the canonical form containing all boundary conditions. This goal has been achieved in the new modified ADM which is called the extended ADM (EADM). The EADM does not require us to calculate the unknown constant which is usually a derivative at the boundary. All numerical approximations by EADM are compared with the results in many other methods such as the modified ADM, FDM, FEM, B-spline, exponential fitting and shooting method. From the results in illustrative examples, it is concluded that EADM is a very effective algorithm which provides promising results with simple calculations. However, it is noted that EADM may encounter difficulties in obtaining each component for the complex nonlinear problems even if symbolic packages are used because each component is obtained by integration.

References

- [1] G. Adomian, *Solving Frontier Problems of Physics: The Decomposition Method*, Kluwer Academic Publications, Dordrecht, 1994.
- [2] G. Adomian, R. Rach, Modified decomposition solution of linear and nonlinear boundary-value problems, *Nonlinear Anal.* 23 (5) (1994) 615–619.
- [3] M. Benabidallah, Y. Cherruault, Application of the Adomian method for solving a class of boundary problems, *Kybernetes* 33 (1) (2004) 118–132.
- [4] H. Caglar, N. Caglar, K. Elfaituri, B-spline interpolation compared with finite difference, finite element and finite volume methods which applied to two-point boundary value problems, *Appl. Math. Comput.* 175 (2006) 72–79.
- [5] Q. Fang, T. Tsuchiya, T. Yamamoto, Finite difference, finite element and finite volume methods applied to two-point boundary value problems, *J. Comput. Appl. Math.* 139 (1) (2002) 9–19.
- [6] S.N. Ha, A nonlinear shooting method for two-point boundary value problems, *Comput. Math. Appl.* 42 (10–11) (2001) 1411–1420.
- [7] M. Inc, D.J. Evans, The decomposition method for solving of a class of singular two-point boundary value problems, *Int. J. Comput. Math.* 80 (7) (2003) 869–882.
- [8] B. Jang, Exact solutions to one dimensional non-homogeneous parabolic problems by the homogeneous Adomian decomposition method, *Appl. Math. Comput.* 186 (2007) 969–979.
- [9] B. Jang, Solutions to the non-homogeneous parabolic problems by the extended HADM, *Appl. Math. Comput.* 191 (2) (2007) 466–483.
- [10] K. Keller, *Numerical Solutions of Two-Point Boundary Value Problems*, SIAM, Philadelphia, 1976.
- [11] O.A. Taiwo, Exponential fitting for the solution of two-point boundary value problems with cubic spline collocation tau-method, *Int. J. Comput. Math.* 79 (3) (2002) 299–306.
- [12] A.M. Wazwaz, A reliable algorithm for obtaining positive solutions for nonlinear boundary value problems, *Comput. Math. Appl.* 41 (2001) 1237–1244.